| Notice of Allowability | Application No. | Applicant(s) | |
|---|---|---|---|
| | 10/666,033 | BATES ET AL. | |
| | Examiner | Art Unit | |
| | Jason Mitchell | 2193 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--*

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *a supplemental amendment filed on 5/30/07*.

2. ☒ The allowed claim(s) is/are *43,52-59,65-75, and 77 (renumbered 1-21)*.

3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
      1. ☐ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
**THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.**

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. ☐ CORRECTED DRAWINGS ( as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review ( PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   **Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).**

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

**Attachment(s)**

1. ☒ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☒ Interview Summary (PTO-413), Paper No./Mail Date *20070606*.
7. ☒ Examiner's Amendment/Comment
8. ☐ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

MENGAL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

## EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Mr. Donald O'Brien on 6/7/07.

**The application has been amended as follows:**

1-42. (Cancelled)

43. (Currently Amended) A computer-implemented method for debugging code, comprising:

while execution is halted at a first point in the code, receiving a user selection of a target call site of the code, the user selection being made through a graphical user interface, and the target call site comprising a call to any one of a plurality of overriding methods; ~~and~~

wherein the code is object-oriented;

in response to the user selection, setting a run-into breakpoint at each respective entry point of the plurality of overriding methods, wherein the run-into breakpoint is configured to halt subsequent execution only when the target call site calls one of the plurality of overriding methods;

while the code is under debug:

encountering one of the plurality of run-into breakpoints, during execution of the code under debug; and

determining whether the target call site called the one of the plurality of overriding methods;

if the target call site called the one of the plurality of overriding methods:

halting execution of the code.

44-51. (Cancelled)

52. (Currently Amended) A computer-implemented method for debugging code, comprising:

(a) while execution is halted at a first point in the code, receiving a user selection of a target call site of the code, the user selection being made through a graphical user interface, and the target call site comprising a call to any one of a plurality of overriding methods;

(b) in response to receiving the user selection, setting a run-into breakpoint at each respective entry point of the plurality of overriding methods;

(c) determining call context information identifying a location of the target call site in the code;

(d) encountering the run-into breakpoint; and

[[(d)]] (e) for each of the run-into breakpoints encountered during execution of the code:

determining whether the target call site called one of the plurality of overriding methods, based on the call context information; and

if so:

halting execution of the code ~~at the target call site~~.

53. (Currently Amended) The method of claim 52, further comprising:

repeating each of the steps (a)-(e) [[(d)]] for a plurality of target call sites, wherein each target call site has an associated method of an associated plurality of overriding methods, an associated breakpoint at each respective associated entry point of the associated plurality of overriding methods, and associated call context information; and

automatically removing at least one of the associated breakpoints upon determining that one of the plurality of target call sites has called the associated method based on the associated call context information.

54. (Previously Presented) The method of claim 52, wherein determining whether the target call site called the one of the plurality of overriding methods comprises comparing the call context information to selected content of a call stack.

55. (Previously Presented) The method of claim 54, wherein the target call site is determined to have called the one of the plurality of overriding methods if stored call context information matches the selected content of the call stack.

56. (Previously Presented) The method of claim 54, wherein the selected content of the call stack is a call to the one of the plurality of overriding methods,

and wherein the target call site is determined to have called the one of the plurality of methods if the stored call context information matches the selected content of the call stack.

57. (Currently Amended) The method of claim 52, wherein the code is [[id]] object-oriented, and the method further comprises identifying the plurality of overriding methods.

58. (Previously Presented) The method of claim 57, further comprising, prior to identifying the plurality of overriding methods, determining that the method is associated with an object.

59. (Previously Presented) The method of claim 57, wherein identifying the plurality of overriding methods comprises traversing a class hierarchy, and locating each matching member method according to the selected target call site.

60-64. (Cancelled)

65. (Currently Amended) A computer readable storage medium containing a program which, when executed, performs an operation for debugging code, comprising:

(a) while execution is halted at a first point in the code, receiving a user selection of a target call site of the code, the user selection being made through a graphical user interface, and the target call site comprising a call to any one of a plurality of overriding methods;

(b) in response to receiving the user selection, setting a run-into breakpoint at each respective entry point of the plurality of overriding methods;

(c) determining call context information identifying a location of the target call site in the code;

(d) encountering the run-into breakpoint; and

[[(d)]] (e) for each of the run-into breakpoints encountered during execution of the code:

determining whether the target call site called one of the plurality of overriding methods, based on the call context information; and

if so:

halting execution of the code ~~at the target call site~~.

66. (Currently Amended) The computer readable storage medium of claim 65, further comprising:

repeating each of the steps (a)-(e) [[(d)]] for a plurality of target call sites, wherein each target call site has an associated method of an associated plurality of overriding methods, an associated breakpoint at each respective associated entry point of the associated plurality of methods, and associated call context information; and

automatically removing at least one of the associated breakpoints upon determining that one of the plurality of target call sites has called the associated method based on the associated call context information.

67. (Previously Presented) The computer readable storage medium of claim 65, wherein determining whether the target call site called the one of the

plurality of overriding methods comprises comparing the call context information to selected content of a call stack.

68. (Previously Presented) The computer readable storage medium of claim 67, wherein the target call site is determined to have called the one of the plurality of overriding methods if stored call context information matches the selected content of the call stack.

69. (Previously Presented) The computer readable storage medium of claim 67, wherein the selected content of the call stack is a call to the one of the plurality of overriding methods, and wherein the target call site is determined to have called the one of the plurality of overriding methods if the stored call context information matches the selected content of the call stack.

70. (Currently Amended) The computer readable storage medium of claim 65, wherein the code is [[id]] object-oriented, and the method further comprises identifying the plurality of overriding methods.

71. (Previously Presented) The computer readable storage medium of claim 70, further comprising, prior to identifying the plurality of overriding methods, determining that the method is associated with an object.

72. (Previously Presented) The computer readable storage medium of claim 70, wherein identifying the plurality of overriding methods comprises traversing a class hierarchy, and locating each matching member method according to the selected target call site.

73. (Currently Amended) A computer, comprising:

a memory;

a processor;

code under debug resident in the memory, the code comprising at least one target call site comprising a call to any one of a plurality of overriding methods, the target call site being selected by the user through a graphical user interface while execution is halted at a first point in the code;

a breakpoint data structure resident in the memory and configured for storing at least context information indicating a location of the target call site within the code; and

a debugger program resident in the memory and which, when executed by the processor, is configured to place a run-into breakpoint at each respective entry point of the plurality of overriding methods in response to the target call site being selected by the user, and is further configured to interrupt execution of the code under debug in response to encountering a breakpoint set on one of the plurality of overriding methods if the selected target call site called the one of the plurality of overriding methods with reference to the context information.

74. (Previously Presented) The system of claim 73, further comprising a caller data structure resident in the memory and configured for storing at least callers of overriding methods in the code as encountered during an execution path; and wherein the debugger program is configured to determine whether the routine is called from the target call site by comparing the context information to a caller stored in the caller data structure.

75. (Previously Presented) The system of claim 73, wherein the code under debug is object-oriented.

76. (Canceled)

77. (New) The method of claim 52 where the call context information unambiguously identifies the location of the target call site in the code.

***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Mitchell whose telephone number is (571) 272-3728. The examiner can normally be reached on Monday-Thursday and alternate Fridays 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Jason Mitchell
6/7/07

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100